Quality Analytics using Google Colab

Table of Contents

Course Objective

This course provides a **hands-on approach** to applying **Quality Analytics** using Python in **Google Colab**. Quality Analytics helps businesses improve **process efficiency, product quality, and defect reduction** through data-driven insights.

By the end of this course, you will:

- Understand **Quality Metrics and Statistical Analysis**
- Use **Python for quality data collection, visualization, and process monitoring**
- Implement **Six Sigma tools** such as Pareto Charts, Control Charts, and Defect Analysis
- Develop **automated Quality Reports**

Introduction to Quality Analytics

Why Quality Analytics?

- The role of **Quality Analytics in manufacturing and services**
- Importance of **data-driven quality management**
- Overview of **Python for statistical process control (SPC) and defect analysis**

Environment Setup

Ensure you have **Google Colab** or a Python 3.x environment with the following:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
```

Quality Data Collection

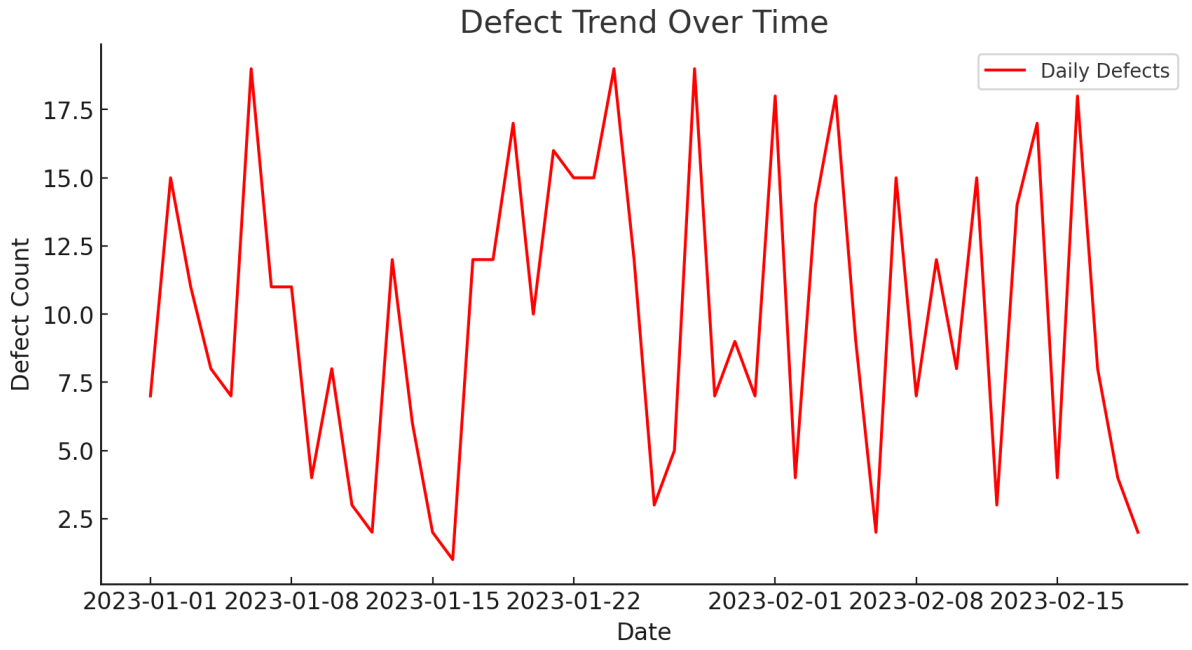
Sample Dataset: Defect Tracking

```
np.random.seed(42)
dates = pd.date_range(start='2023-01-01', periods=50, freq='D')
defects = np.random.randint(1, 20, size=50)

defect_df = pd.DataFrame({'Date': dates, 'Defect Count':
defects})
defect_df.set_index('Date', inplace=True)
defect_df.head()
```

Data Visualization: Defect Trends

```
plt.figure(figsize=(10,5))
sns.lineplot(x=defect_df.index, y=defect_df['Defect Count'],
color='red', label='Daily Defects')
plt.title('Defect Trend Over Time')
plt.xlabel('Date')
plt.ylabel('Defect Count')
plt.legend()
plt.grid()
plt.show()
```



Pareto Analysis

Identifying Key Defect Types

```
defect_types = ['Scratch', 'Dent', 'Crack', 'Misalignment',
'Color Defect']
defect_counts = [120, 95, 60, 30, 20]

df_pareto = pd.DataFrame({'Defect Type': defect_types, 'Count':
defect_counts})
df_pareto.sort_values(by='Count', ascending=False, inplace=True)
df_pareto['Cumulative %'] = df_pareto['Count'].cumsum() /
df_pareto['Count'].sum() * 100
df_pareto
```

Visualizing Pareto Chart

```
fig, ax1 = plt.subplots(figsize=(8,5))
ax2 = ax1.twinx()

ax1.bar(df_pareto['Defect Type'], df_pareto['Count'],
color='blue', label='Defect Count')
ax2.plot(df_pareto['Defect Type'], df_pareto['Cumulative %'],
color='red', marker='o', linestyle='-', label='Cumulative %')

ax1.set_xlabel('Defect Type')
ax1.set_ylabel('Defect Count', color='blue')
ax2.set_ylabel('Cumulative %', color='red')
ax1.set_title('Pareto Analysis of Defects')
ax1.grid()
fig.legend()
plt.show()
```

Control Charts

Computing Control Limits

```
mean_defects = defect_df['Defect Count'].mean()
sigma = defect_df['Defect Count'].std()
UCL = mean_defects + (3 * sigma)
LCL = max(0, mean_defects - (3 * sigma))  # LCL can't be
negative
```

Plotting Control Chart

```
plt.figure(figsize=(10,5))
plt.plot(defect_df.index, defect_df['Defect Count'], marker='o',
linestyle='-', label='Defect Count')
plt.axhline(UCL, color='red', linestyle='--', label='UCL')
plt.axhline(mean_defects, color='green', linestyle='-',
label='Center Line')
plt.axhline(LCL, color='red', linestyle='--', label='LCL')
plt.title('Control Chart for Quality Monitoring')
plt.xlabel('Date')
plt.ylabel('Defect Count')
plt.legend()
plt.grid()
plt.show()
```

Process Capability Analysis

Calculating Cp & Cpk

```
USL, LSL = 25, 5  # Upper and Lower Specification Limits
Cp = (USL - LSL) / (6 * sigma)
Cpk = min((USL - mean_defects) / (3 * sigma), (mean_defects -
LSL) / (3 * sigma))

print(f"Process Capability Index (Cp): {Cp:.2f}")
print(f"Process Capability Performance (Cpk): {Cpk:.2f}")
```

Visualizing Quality Distribution

```
plt.figure(figsize=(10,5))
sns.histplot(defect_df['Defect Count'], bins=10, kde=True,
color='purple', label='Defect Data')
plt.axvline(USL, color='black', linestyle='--', label='USL')
plt.axvline(LSL, color='black', linestyle='--', label='LSL')
plt.axvline(mean_defects, color='green', linestyle='-',
label='Mean')
plt.title('Quality Distribution & Process Capability')
plt.xlabel('Defect Count')
plt.ylabel('Frequency')
plt.legend()
plt.show()
```

Course Summary

- Analyzed Quality Data using Python for Process Improvement
- Generated and Visualized Defect Trends for Monitoring
- Implemented Pareto Charts for Key Defect Analysis
- Developed Control Charts and Conducted Process Capability Analysis